



Matragi 6-3

AP #  
JW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

5

Applicant(s): Matragi et al.

Case: 6-3

Serial No.: 09/488,181

Filing Date: January 20, 2000

10

Group: 2666

Examiner: Ronald B. Abelson

I hereby certify that this paper is being deposited on this date with the U.S. Postal Service as first class mail addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Signature: *Babette Blak* Date: April 21, 2005

Title: Method and Apparatus for Message-based Overload Control in a Distributed Call-Processor Communication System

15

APPEAL BRIEF

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

20 Sir:

Applicants hereby appeal the final rejection dated December 29, 2004, of claims 1 through 24 of the above-identified patent application.

25

REAL PARTY IN INTEREST

The present application is assigned to Lucent Technologies Inc., as evidenced by an assignment recorded on January 20, 2000 in the United States Patent and Trademark Office at Reel 010562, Frame 0202. The assignee, Lucent Technologies Inc., is the real party in interest.

30

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

04/26/2005 MAHME1 00000004 500762 09488181

01 FC:1402 500.00 DA

### STATUS OF CLAIMS

Claims 1 through 24 are presently pending in the above-identified patent application. Claims 1-4, 6-9, 12-16, 18-21, and 24 are rejected under 35 U.S.C. §103(a) as being unpatentable over McAllister et al. (United States Patent Number 6,215,765) in view of Kunimoto et al. (United States Patent Number 6,396,808). The Examiner indicated that claims 5, 10, 11, 17, 22, and 23 would be allowable if rewritten in independent form including all of the limitations of the base claims and any intervening claims.

### STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection.

### SUMMARY OF CLAIMED SUBJECT MATTER

The present invention is directed to a method and apparatus for alleviating congestion and overload in a distributed call-processing system interconnected through a packet based network, such as an IP or an ATM network. The illustrative IP network includes a plurality of end terminals (ETs) and distributed call processors (CPs) (page 4, line 24, to page 5, line 6). When an end terminal (ET) wants to place a call, the end terminal (ET) send a call set up message to a call processor (CP). According to an aspect of the invention, the call processor will determine whether to process the request or to forward the request to another call processor (page 5, lines 7-17). Generally, the call processor will declare an overload condition if sufficient resources (such as processing or memory resources) are not available to process a given call. If a call processor determines that it is too congested to process a call, the call processor enters an overload condition, selects an alternate call processor and forwards the request to the alternate call processor (page 5, line 18, to page 6, line 10).

### STATEMENT OF GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-4, 6-9, 12-16, 18-21, and 24 are rejected under 35 U.S.C. §103(a) as being unpatentable over McAllister et al. in view of Kunimoto et al.

ARGUMENTClaims 1, 8, 12, 13, 20 and 24

Claims 1, 8, 12, 13, 20, and 24 are rejected under 35 U.S.C. §103(a) as being unpatentable over McAllister et al. in view of Kunimoto et al.

5           Regarding claims 1, 8, 13, and 20, the Examiner asserts that McAllister discloses that “the system identifies an alternate call processor to process the call set up request using a list of call processors if sufficient resources do not exist (col. 3, lines 32-37).” The Examiner also asserts that Kunimoto teaches “forwarding the call set up request to the identified alternate call processor with an identifier of said congested call processor” (FIG. 10   24, box 40A; col. 23, line 66, to col. 24, line 6). The Examiner asserts that, therefore, it would have been obvious to one of ordinary skill in the art, to modify the system of McAllister according to the teachings of Kunimoto. Regarding claims 12 and 24, the Examiner asserts that McAllister discloses that the system receives a call set up request from an end terminal (col. 3, lines 5-7), determines if sufficient resources exist to process the call 15   set up request (col. 3, lines 7-9), and identifies an alternate call processor to process the call set up request (col. 3, lines 25-27) using the flag associated with each potential alternate call processor (col. 3, lines 32-37).

Applicants note that McAllister is directed to rerouting a call due to congestion or physical failure. (See, Abstract.) McAllister defines congestion in regard to 20   network links, not call processors. McAllister teaches that “congestion may occur on a network link if many incoming streams of traffic all terminate on the same outbound link, or the outbound link may (be) busy or down due to failure.” (Col. 1, lines 10-12.) Applicants therefore maintain that McAllister is addressing the congestion on a network link and not a call processor. In particular, McAllister teaches that, if the bandwidth of “many incoming 25   streams of traffic” exceeds the bandwidth of the “same outbound link,” then the outbound link will be congested. The call processor, however, will *not* be congested if it has enough processing power to handle the bandwidth of the incoming streams of traffic. Thus, network link congestion is *not* the same as call processor congestion.

In addition, since McAllister does not address call processors or the 30   congestion of call processors, McAllister does *not* disclose or suggest a *list of call*

processors, and does *not* disclose or suggest a list of call processors that includes a *congestion status* of one or more of said call processors or *setting a flag associated with said congested call processor* indicating that said congested call processor is congested.

Applicants also note that Kunimoto teaches that a call processor has a  
 5 *designated* back-up processor (ATM switching system 2B; col. 23, line 66, to col. 24, line 6).

The present invention is directed to a *distributed* method for identifying the “back-up” call processor, wherein one or more call processors maintain a *list that contains the congestion status of one or more of the call processors*. The list is utilized by a call processor to identify one or more back-up call processors. Independent claims 1 and 13 require *identifying* an  
 10 alternate call processor to process said call set up request *using a list of call processors* if sufficient resources do not exist, “wherein said list of call processors includes a *congestion status* of one or more of said call processors.” Kunimoto does *not* disclose or suggest a *list of call processors* that includes a *congestion status* of one or more call processors, and does not disclose or suggest *identifying* an alternate call processor to process said call set up  
 15 request *using a list of call processors*.

Independent claims 8 and 20 require setting a flag associated with said congested call processor indicating that said congested call processor is congested by  
 utilizing *said received call set up request*. Thus, the flag is set in the *call processor that receives the forwarded call set up request* that indicates congestion. Kunimoto does *not*  
 20 disclose or suggest *setting a flag associated with a congested call processor* indicating that the congested call processor is congested by *utilizing a received call set up request*.

Independent claims 12 and 24 require *identifying an alternate call processor* to process said call set up request using said *flag associated with each potential alternate call processor*. Contrary to the Examiner’s assertion (and as noted above), neither McAllister nor  
 25 Kunimoto disclose or suggest this limitation.

Thus, McAllister et al. and Kunimoto et al., alone or in combination, do not disclose or suggest a list of call processors that includes a congestion status of one or more of said call processors, as required by independent claims 1 and 13, do not disclose or suggest  
 30 identifying an alternate call processor to process said call set up request using a list of call processors if sufficient resources do not exist, as required by independent claims 1 and 13, do

not disclose or suggest setting a flag associated with said congested call processor indicating that said congested call processor is congested by utilizing said received call set up request, as required by independent claims 8 and 20, and do not disclose or suggest identifying an alternate call processor to process said call set up request using said flag associated with each potential alternate call processor, as required by claims 12 and 24.

#### Claims 2 and 14

Regarding claims 2 and 14, the Examiner asserts that McAllister discloses that the call processor that previously received a forwarded call set up request within a predefined interval is not selected as the alternate call processor during the identifying step (col. 3, lines 34-37). Applicants note that McAllister does not define a *predefined interval*, and does not disclose or suggest a predefined interval during which a call processor that previously received a forwarded call set up request will not be selected as the alternate call processor during said identifying step.

Thus, McAllister et al. and Kunimoto et al., alone or in combination, do not disclose or suggest wherein a call processor that previously received a forwarded call set up request within a predefined interval is not selected as the alternate call processor during said identifying step, as required by claims 2 and 14.

#### Claims 6 and 18

Regarding claims 6 and 18, the Examiner asserts that McAllister discloses evaluating a total congestion indicator flag indicating whether all potential alternate call processors are congested (col. 3, lines 39-41). Contrary to the Examiner's assertion, Applicants note that neither McAllister nor Kunimoto disclose or suggest a *total congestion indicator flag* that indicates whether *all potential alternate call processors are congested*, and do not disclose or suggest *evaluating* such a flag.

Thus, McAllister et al. and Kunimoto et al., alone or in combination, do not disclose or suggest wherein said identifying step further comprises the step of evaluating a total congestion indicator flag indicating whether all potential alternate call processors are congested, as required by claims 6 and 18.

Conclusion

The rejections of the cited claims under section 103 in view of McAllister et al. and Kunimoto et al., alone or in any combination, are therefore believed to be improper and should be withdrawn. The remaining rejected dependent claims are believed allowable  
5 for at least the reasons identified above with respect to the independent claims.

The attention of the Examiner and the Appeal Board to this matter is appreciated.

Respectfully,



Kevin M. Mason

Attorney for Applicant(s)

Reg. No. 36,597

Ryan, Mason & Lewis, LLP

1300 Post Road, Suite 205

Fairfield, CT 06824

(203) 255-6560

10 Date: April 21, 2005

15

APPENDIX

1. An overload control method for use in a network employing distributed call-processing, said method comprising the steps of:
  - 5 receiving a call set up request from an end terminal;  
determining if sufficient resources exist in a call processor to process said call set up request;  
identifying an alternate call processor to process said call set up request using a list of call processors if sufficient resources do not exist, wherein said list of call processors includes a  
10 congestion status of one or more of said call processors; and  
forwarding said call set up request to said identified alternate call processor with an identifier of said congested call processor, whereby said forwarded call set up request indicates to said alternate call processor that said congested call processor is congested.
- 15 2. The method of claim 1, wherein a call processor that previously received a forwarded call set up request within a predefined interval is not selected as the alternate call processor during said identifying step.
- 20 3. The method of claim 1, wherein said identifying step further comprises the step of evaluating a congestion indicator flag associated with each potential alternate call processor, wherein said congestion indicator flag is set if a congestion message is received from said corresponding alternate call processor.
- 25 4. The method of claim 1, wherein said forwarding step further comprises the step of setting a flag indicating that said selected alternate call processor received said forwarded call set up request.
- 30 5. The method of claim 4, wherein said flag indicating that said selected alternate call processor received said forwarded call set up request automatically expires after a predefined interval.

6. The method of claim 1, wherein said identifying step further comprises the step of evaluating a total congestion indicator flag indicating whether all potential alternate call processors are congested.

5 7. The method of claim 1, wherein said list of call processors is an ordered list.

8. An overload control method for use in a network employing distributed call-processing, said method comprising the steps of:

10 receiving a forwarded call set up request from a congested call processor, said forwarded call set up request including an identifier of said congested call processor; and setting a flag associated with said congested call processor indicating that said congested call processor is congested by utilizing said received call set up request.

15 9. The method of claim 8, further comprising the step of determining if sufficient resources exist to process said forwarded call set up request.

10. The method of claim 8, further comprising the step of setting a timer associated with said flag.

20 11. The method of claim 10, further comprising the step of automatically expiring said flag in accordance with said timer.

25 12. The method of claim 8, further comprising the steps of receiving a call set up request from an end terminal, determining if sufficient resources exist to process said call set up request and identifying an alternate call processor to process said call set up request using said flag associated with each potential alternate call processor.



13. An overload control manager for use in a network employing distributed call-processing, comprising:

a memory for storing computer readable code; and

a processor operatively coupled to said memory, said processor configured to:

5 receive a call set up request from an end terminal;

determine if sufficient resources exist in a call processor to process said call set up request;

identify an alternate call processor to process said call set up request using a list of call processors if sufficient resources do not exist, wherein said list of call processors includes a

10 congestion status of one or more of said call processors; and

forward said call set up request to said identified alternate call processor with an identifier of said congested call processor, whereby said forwarded call set up request indicates to said alternate call processor that said congested call processor is congested.

15 14. The overload control manager of claim 13, wherein a call processor that previously received a forwarded call set up request within a predefined interval is not selected as the alternate call processor during said identifying step.

20 15. The overload control manager of claim 13, wherein said processor is further configured to evaluate a congestion indicator flag associated with each potential alternate call processor, wherein said congestion indicator flag is set if a congestion message is received from said corresponding alternate call processor.

25 16. The overload control manager of claim 13, wherein said processor is further configured to set a flag indicating that said selected alternate call processor received said forwarded call set up request.

17. The overload control manager of claim 16, wherein said flag indicating that said selected alternate call processor received said forwarded call set up request automatically expires after a predefined interval.

5 18. The overload control manager of claim 13, wherein said processor is further configured to evaluate a total congestion indicator flag indicating whether all potential alternate call processors are congested.

10 19. The overload control manager of claim 13, wherein said list of call processors is an ordered list.

20. An overload control manager for use in a network employing distributed call-processing, comprising:

15 a memory for storing computer readable code; and  
a processor operatively coupled to said memory, said processor configured to:  
receiving a forwarded call set up request from a congested call processor, said forwarded call set up request including an identifier of said congested call processor; and  
setting a flag associated with said congested call processor indicating that said congested call processor is congested by utilizing said received call set up request.

20

21. The overload control manager of claim 20, wherein said processor is further configured to determine if sufficient resources exist to process said forwarded call set up request.

25 22. The overload control manager of claim 20, wherein said processor is further configured to set a timer associated with said flag.

23. The overload control manager of claim 20, wherein said processor is further configured to automatically expire said flag in accordance with said timer.

24. The overload control manager of claim 20, wherein said processor is further configured to (i) receive a call set up request from an end terminal, (ii) determine if sufficient resources exist to process said call set up request and (iii) identify an alternate call processor to process said call set up request using said flag associated with each potential alternate call processor.



Matragi 6-3

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): Matragi et al.  
Case: 6-3  
Serial No.: 09/488,181  
Filing Date: January 20, 2000  
Group: 2666  
Title: Method and Apparatus For Message-Based Overload Control in a Distributed Call-Processor Communication System

I hereby certify that this paper is being deposited on this date with the U.S. Postal Service as first class mail addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Signature: Bobbett Blake Date: April 21, 2005

TRANSMITTAL OF APPEAL BRIEF

Mail Stop Appeal Brief Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

Submitted herewith are the following documents relating to the above-identified patent application:

- (1) Appeal Brief; and
- (2) Copy of Notice of Appeal, filed on February 17, 2005, with copy of stamped return postcard indicating receipt of Notice by PTO on February 22, 2005.

There is an additional fee of \$500 due in conjunction with this submission under 37 CFR §1.17(c). Please charge **Deposit Account No. 50-0762** the amount of \$500, to cover this fee. In the event of non-payment or improper payment of a required fee, the Commissioner is authorized to charge or to credit **Deposit Account No. 50-0762** as required to correct the error. A duplicate copy of this letter is enclosed.

Respectfully,

Kevin M. Mason

Date: April 21, 2005

Kevin M. Mason  
Attorney for Applicant(s)  
Reg. No. 36,597  
Ryan, Mason & Lewis, LLP  
1300 Post Road, Suite 205  
Fairfield, CT 06824  
(203) 255-6560